Bases de Données II, 29 mai 2015

NOM + PRÉNOM :

Orientation + Année :

Cet examen contient 13 questions. Durée : 3 heures et 30 minutes.

A written English translation of the questions is available upon request.

Un catalogue de livres est stocké dans un document XML. Les prix sont affichés en EUR ou USD, et peuvent varier d'un vendeur à l'autre.

La DTD est incluse au début du document XML de la figure 1. Le livre "XML Developer's Guide" coûte 44.95 USD auprès d'Amazon.com, et 34.00 EUR auprès de Proxis.com. **Le cours de change en vigueur est de 1 EUR = 1.34 USD.** Puisque 34.00 EUR = 45.56 USD, Amazon.com offre un meilleur prix que Proxis.com. L'auteur de ce livre s'appelle Matthew Gambardella.

Pour les questions 1 à 4 évitez, si possible, l'usage des axes suivants : parent, ancestor, following-sibling, preceding-sibling, following et preceding.

Question 1 Écrivez une expression XPath (aussi simple que possible) qui rend chaque nœud de type `texte` dont la valeur est le titre d'un livre qui n'est pas disponible à un prix inférieur à 10.00 EUR (ou 13.40 USD ; le cours de change en vigueur est de 1 EUR = 1.34 USD). Pour le document de la figure 1, la réponse consiste en `XML Developer's Guide`, `Microsoft .NET: The Programming Bible` et `MSXML3: A Comprehensive Guide`.

.../5

Question 2 Écrivez une expression XPath (aussi simple que possible) qui rend chaque nœud de type `texte` dont la valeur est le nom de famille d'un auteur ayant écrit au moins deux livres. Essayez d'éviter d'afficher un même nom plusieurs fois. Pour le document de la figure 1, la réponse consiste en `Corets` et `O'Brien`.

.../5

Question 3 Écrivez une expression XPath (aussi simple que possible) qui rend chaque nœud de type `texte` dont la valeur est le titre d'un livre qui est écrit par l'auteur du livre intitulé "The Sundered Grail". Pour le document de la figure 1, la réponse consiste en `Maeve Ascendant`, `Oberon's Legacy` et `The Sundered Grail`. Notez que l'expression ne peut pas contenir des constantes différentes de `The Sundered Grail`.

.../5

Question 4 Écrivez une expression XPath (aussi simple que possible) qui rend chaque nœud de type `texte` dont la valeur est le nom de famille d'un auteur ayant écrit des livres dans deux ou plusieurs genres différents, parmi lesquels se trouve le genre "Fantasy". Notez que l'expression ne peut pas contenir des constantes différentes de `Fantasy`. Le même nom de famille peut apparaitre plusieurs fois ; pas besoin d'éliminer les doublons. Pour le document de la figure 1, la réponse consiste en `Corets`.

.../5

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE catalog [
<!ELEMENT catalog (book*)>
<!ELEMENT book (author, title, genre, sales)>
<!ELEMENT author (last, first)>
<!ELEMENT last (#PCDATA)>
<!ELEMENT first (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT genre (#PCDATA)>
<!ELEMENT sales (sale*)>
<!ELEMENT sale (#PCDATA)>
<!ATTLIST book id CDATA #REQUIRED>
<!ATTLIST sale bookstore CDATA #REQUIRED>
<!ATTLIST sale unit (EUR|USD) #REQUIRED>
]>
<catalog>
  <book id="bk101">
    <author><last>Gambardella</last><first>Matthew</first></author>
    <title>XML Developer's Guide</title><genre>Computer</genre>
    <sales><sale bookstore="Amazon.com" unit="USD">44.95</sale>
           <sale bookstore="Proxis.com" unit="EUR">34.00</sale>
           <sale bookstore="AbeBooks" unit="EUR">33.50</sale>
    </sales>
  </book>
  <book id="bk103">
    <author><last>Corets</last><first>Eva</first></author>
    <title>Maeve Ascendant</title><genre>Science Fiction</genre>
    <sales><sale bookstore="Biblio.com" unit="USD">11.95</sale>
           <sale bookstore="Proxis.com" unit="EUR">12.00</sale>
    </sales>
  </book>
  <book id="bk104">
    <author><last>Corets</last><first>Eva</first></author>
    <title>Oberon's Legacy</title><genre>Fantasy</genre>
    <sales><sale bookstore="Alibris" unit="EUR">11.95</sale>
           <sale bookstore="Proxis.com" unit="EUR">9.00</sale>
    </sales>
  </book>
  <book id="bk105">
    <author><last>Corets</last><first>Eva</first></author>
    <title>The Sundered Grail</title><genre>Fantasy</genre>
    <sales><sale bookstore="Amazon.com" unit="USD">5.95</sale>
           <sale bookstore="Books-A-Million" unit="USD">5.95</sale>
    </sales>
  </book>
  <book id="bk109">
    <author><last>Kress</last><first>Peter</first></author>
    <title>Paradox Lost</title><genre>Science Fiction</genre>
    <sales><sale bookstore="Amazon.com" unit="USD">14.95</sale>
           <sale bookstore="Proxis.com" unit="EUR">9.95</sale>
    </sales>
  </book>
  <book id="bk110">
    <author><last>O'Brien</last><first>Tim</first></author>
    <title>Microsoft .NET: The Programming Bible</title><genre>Computer</genre>
    <sales><sale bookstore="Biblio.com" unit="USD">36.95</sale>
           <sale bookstore="Proxis.com" unit="EUR">27.00</sale>
    </sales>
  </book>
  <book id="bk111">
    <author><last>O'Brien</last><first>Tim</first></author>
    <title>MSXML3: A Comprehensive Guide</title><genre>Computer</genre>
    <sales><sale bookstore="Amazon.com" unit="USD">36.95</sale>
           <sale bookstore="Alibris" unit="EUR">27.00</sale>
    </sales>
  </book>
</catalog>
```

FIGURE 1 – Catalogue de livres.

Question 5 Écrivez un programme XQuery qui retourne une liste de livres dont les prix ne sont listés qu'en euros. Retournez le titre du livre dans une balise `title`, avec un attribut `minprice` mentionnant son prix le plus bas parmi ses ventes. La sortie devrait être

```
<title minprice="9">Oberon's Legacy</title>
```

.../5

Question 6 Écrivez un programme XQuery qui retourne la liste des livres classés par genre dans le format illustré par la figure 2. Les genres sont classés par ordre alphabétique, et les livres, par le nombre de vendeurs qui les offrent.

.../10

```
<genre name="Computer">
  <book id="bk110">Microsoft .NET: The Programming Bible</book>
  <book id="bk111">MSXML3: A Comprehensive Guide</book>
  <book id="bk101">XML Developer's Guide</book>
</genre>
<genre name="Fantasy">
  <book id="bk104">Oberon's Legacy</book>
  <book id="bk105">The Sundered Grail</book>
</genre>
<genre name="Science Fiction">
  <book id="bk103">Maeve Ascendant</book>
  <book id="bk109">Paradox Lost</book>
</genre>
```

FIGURE 2 – Output du programme XQuery de la question 6.

Question 7 Écrivez un programme XSLT qui génère un document XML affichant les titres de livre par vendeur, dans le format illustré par la figure 3. Tous les prix sont affichés en EUR, en utilisant le cours de change de 1 EUR = 1.34 USD.

La position des blancs et retours à la ligne n'a pas d'importance. Le programme ne peut pas contenir des `xsl:for-each` ou `xsl:if`.

```
<bookstores>
  <bookstore name="Amazon.com">
    <title Euros="33.54">XML Developer's Guide</title>
    <title Euros="4.44">The Sundered Grail</title>
    <title Euros="11.16">Paradox Lost</title>
    <title Euros="27.57">MSXML3: A Comprehensive Guide</title>
  </bookstore>
  <bookstore name="Proxis.com">
    <title Euros="34.00">XML Developer's Guide</title>
    <title Euros="12.00">Maeve Ascendant</title>
    <title Euros="9.00">Oberon's Legacy</title>
    <title Euros="9.95">Paradox Lost</title>
    <title Euros="27.00">Microsoft .NET: The Programming Bible</title>
  </bookstore>
  <bookstore name="AbeBooks">
    <title Euros="33.50">XML Developer's Guide</title>
  </bookstore>
  <bookstore name="Biblio.com">
    <title Euros="8.92">Maeve Ascendant</title>
    <title Euros="27.57">Microsoft .NET: The Programming Bible</title>
  </bookstore>
  <bookstore name="Alibris">
    <title Euros="11.95">Oberon's Legacy</title>
    <title Euros="27.00">MSXML3: A Comprehensive Guide</title>
  </bookstore>
  <bookstore name="Books-A-Million">
    <title Euros="4.44">The Sundered Grail</title>
  </bookstore>
</bookstores>
```

FIGURE 3 – Output du programme XSLT.

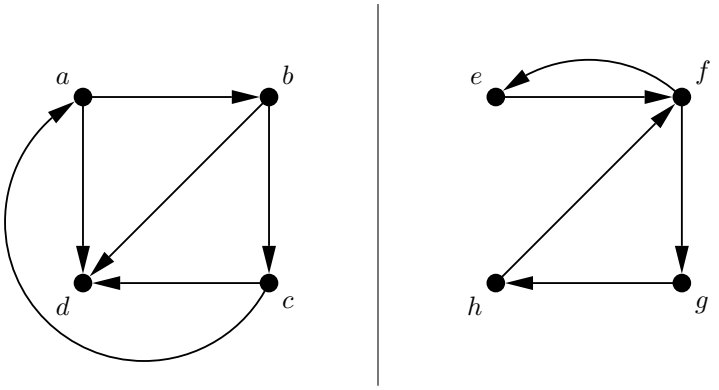

FIGURE 4 – Deux graphes dirigés.

Un graphe dirigé est encodé en utilisant le prédicat E pour les arêtes. On peut supposer que chaque nœud a une arête entrante ou sortante. Soient a, b deux nœuds. Soit $k \geq 1$. Un *chemin dirigé de a à b* est une séquence de nœuds $\langle n_1, n_2, \ldots, n_k \rangle$ telle que $a = n_1$, $b = n_k$ et pour tout $i \in \{1, \ldots, k-1\}$, $E(n_i, n_{i+1})$ est une arête du graphe. La *longueur* de ce chemin est $k - 1$. Un graphe dirigé est *fortement connexe* si pour tous les nœuds a et b, il existe un chemin dirigé de a à b et un chemin dirigé de b à a. Un nœud a d'un graphe dirigé est *un centre* si pour tout nœud b, il existe un chemin dirigé de b à a dont la longueur est strictement inférieure à 3.

Par exemple, le graphe de gauche dans la figure 4 est encodé par $\{E(a,b), E(b,c), E(b,d), E(c,d), E(a,d), E(c,a)\}$. Dans ce graphe, $\langle a, b, c, d \rangle$ est un chemin dirigé de a à d, de longueur 3. Le graphe n'est pas fortement connexe, car il n'y a pas de chemin de d à a. Le seul centre est d. Le graphe de droite dans la figure 4 est fortement connexe. Les centres sont f et g. Le nœud h n'est pas un centre car le plus court chemin de e à h est de longueur 3.

Question 8 Écrivez un programme en datalog$^\neg$ (i.e., datalog avec négation) pour le prédicat idb *Centre*(x) qui signifie que x est un centre du graphe.

.../10

Question 9 Écrivez un programme en datalog$^\neg$ (i.e., datalog avec négation) pour le prédicat idb Fc(“oui”) qui vérifie si un graphe est fortement connexe.

Ne soyez pas perturbé : un programme datalog$^\neg$ peut avoir “oui” comme réponse. Par exemple,

$$Cycle(\text{“oui”}) \leftarrow E(x, y), E(y, x)$$

retourne $Cycle$(“oui”) si et seulement si le graphe contient un cycle de taille 2.

.../10

Question 10 Voici une requête qui est l'union de trois requêtes conjonctives. Simplifiez cette requête en s'appuyant sur les théorèmes vus au cours ou expliquez pourquoi aucune simplification n'est possible.

$$\begin{cases} Answer(x,y) & \leftarrow & R(x,v,v), R(x,v,w), R(u,u,y), R(v,w,u), R(w,u,y) \\ Answer(y,x) & \leftarrow & R(y,u,u), R(u,v,w), R(w,w,x) \\ Answer(y,x) & \leftarrow & R(y,u,v), R(v,v,v), R(v,w,x) \end{cases}$$

.../10

Question 11 On considère les trois relations edb suivantes :
– $\mathit{Frequente}(x, y)$ signifie que x est un buveur qui fréquente le bar y. Par exemple, $\mathit{Frequente}(\text{Ed}, \text{Calypso})$.
– $\mathit{Sert}(y, z)$ signifie que y un bar qui sert la bière z. Par exemple, $\mathit{Sert}(\text{Calypso}, \text{Orval})$.
– $\mathit{Aime}(x, z)$ signifie que x est un buveur qui aime la bière z. Par exemple, $\mathit{Aime}(\text{Ed}, \text{Orval})$.
On suppose que chaque buveur aime au moins une bière et fréquente au moins un bar, que chaque bar sert au moins une bière et que chaque bière est servie dans au moins un bar. Exprimer en datalog$^\neg$ le prédicat idb $\mathit{Attentif}(y)$ qui signifie que y est un bar tel que chaque buveur qui fréquente y aime au moins une bière servie par y.

.../10

Question 12 Dessinez le *dependency graph* pour le programme datalog$^\neg$ suivant. Donnez le résultat de ce programme pour la base de données $\{E(a,b), E(b,c), E(c,d), F(a,d), F(b,a), F(c,b)\}$. Détaillez les calculs.

$$\begin{cases} R(x,y) & \leftarrow & E(x,y) \\ G(x,y) & \leftarrow & F(x,y), \neg R(x,y) \\ G(x,z) & \leftarrow & G(x,y), F(y,z) \\ R(x,z) & \leftarrow & R(x,y), R(y,z) \\ B(x,y) & \leftarrow & F(x,y), \neg G(x,y) \end{cases}$$

.../10

Question 13 Soit **S** le schéma $\mathbf{S} = \{R[B, C, E], S[A, B], T[A, D], U[D, F], V[A, B, C, D]\}$. Supposons que l'on souhaite calculer la jointure de ces cinq relations. Puisque l'opérateur $\bowtie$ est commutatif et associatif, le résultat final ne dépend pas de l'ordre dans lequel on effectue les opérations. Cependant, l'ordre peut être important d'un point de vue pratique.

1. Illustrez à l'aide d'une base de données concrète pourquoi l'expression suivante pose un problème pratique, même si aucun tuple de la base n'est *dangling* :

$$((((R \bowtie S) \bowtie T) \bowtie U) \bowtie V)\,.$$

   Notez que les parenthèses fixent l'ordre dans lequel on effectue les opérations.

2. Donnez une expression (un ordre) qui évite ce problème. Expliquez comment vous trouvez cette expression.

.../10